

CLAIMS

What is claimed is:

1. In at least one electronic device, a method of communicating between a first computer aided design (CAD) application and a second CAD application, comprising:
  - storing native data and a sub-set of native data;
  - providing a plug-in having an application program interface (API) and being accessible by the second CAD application; and
  - the plug-in conveying the sub-set of native data to second CAD application.
2. The method of claim 1, wherein storing comprises placing the native data and the sub-set of native data on a recordable medium.
3. The method of claim 1, wherein the native data comprises data forming a model of an object in the first CAD application.
4. The method of claim 1, wherein the sub-set of native data results from processing the native data with at least one routine from a first library of executable routines to derive the sub-set of native data.
5. The method of claim 4, wherein the first library of executable routines is embedded within the first CAD program.
6. The method of claim 4, wherein the first library of executable routines is a component accessible by the first CAD program.
7. The method of claim 1, wherein the plug-in comprises a second library of executable routines.

8. The method of claim 1, wherein conveying further comprises the second CAD application calling a second library of executable routines to utilize an API to request the native data and the sub-set of native data.

5 9. The method of claim 1, further comprising providing a first library of executable routines.

10. The method of claim 1, further comprising generating the native data as an object is modeled in the first CAD application.

10 11. The method of claim 1, further comprising creating the sub-set of native data utilizing a first library of executable routines.

12. The method of claim 1, further comprising providing a third library of executable routines accessible by the second CAD application.

13. The method of claim 12, wherein the third library of executable routines is embedded within the second CAD application.

14. The method of claim 12, wherein the third library of executable routines is a component accessible by the second CAD application.

15. The method of claim 1, further comprising the first CAD application notifying the second CAD application about updates to the sub-set of native data.

25 16. The method of claim 1, further comprising the first CAD application receiving a notification of a modification of the object from the second CAD application.

17. The method of claim 1, further comprising at least one of the first CAD application and the second CAD application communicating with an analysis application.

30 18. The method of claim 1, further comprising at least one of the first CAD application and the second CAD application communicating with a manufacturing application.

19. In at least one electronic device, a method of communicating between a first CAD application and a second CAD application, comprising:

providing a model of an object on the first CAD application;

utilizing a plug-in in the second CAD application to retrieve a sub-set of native data from the first CAD application relating to the model of the object;

wherein the sub-set of native data enables the second CAD application to create a second model of at least a portion of the object on the second CAD application.

20. The method of claim 19, wherein native data and the sub-set of native data are stored on a recordable medium.

21. The method of claim 19, further comprising providing native data in the form of data relating to the model of the object in the first CAD application.

22. The method of claim 21, wherein the sub-set of native data results from processing the native data with at least one routine from a first library of executable routines to derive the sub-set of native data.

23. The method of claim 22, wherein the first library of executable routines is embedded within the first CAD program.

24. The method of claim 22, wherein the first library of executable routines is a component accessible by the first CAD program.

25. The method of claim 19, wherein the plug-in comprises a second library of executable routines.

26. The method of claim 25, wherein utilizing the plug-in further comprises calling the second library of executable routines to utilize an API to retrieve native data and the sub-set of native data.

27. The method of claim 26, wherein utilizing the plug-in further comprises filtering the native data to extract the sub-set of native data.

28. The method of claim 27, wherein utilizing the plug-in further comprises receiving the sub-set of native data from the first CAD application.

29. The method of claim 25, wherein utilizing the plug-in further comprises calling the second library of executable routines to utilize an API to retrieve the sub-set of native data.

30. The method of claim 29, wherein utilizing the plug-in further comprises receiving the sub-set of native data from the first CAD application.

31. The method of claim 19, further comprising providing a third library of executable routines accessible by the second CAD application.

32. The method of claim 31, wherein the third library of executable routines is embedded within the second CAD application.

33. The method of claim 31, wherein the third library of executable routines is a component of the second CAD application.

34. The method of claim 31, further comprising reading the sub-set of native data utilizing the third library of executable routines to translate the sub-set of native data and create the second model of at least a portion of the object in the second CAD application.

35. The method of claim 19, further comprising the second CAD application receiving notifications from the first CAD application about updates to the sub-set of native data.

36. The method of claim 19, wherein the sub-set of native data includes a complete history of the object in a manner enabling the second CAD application to review the history and modify the object.

37. The method of claim 19, further comprising communicating a modification of the object in the second CAD application to the first CAD application.

38. A system for modeling an object, comprising:

- 5           a first CAD application;  
          a second CAD application;  
          native data and a sub-set of native data relating to an object modeled on the first CAD application stored in a first memory store;  
          a plug-in accessible by the second CAD application and suitable for accessing and  
10       retrieving the sub-set of native data to enable the second CAD application to create a second model of at least a portion of the object modeled on the first CAD application without the first CAD application having to export a file containing the object.

39. The system of claim 38, further comprising a first library of executable routines accessible by the first CAD application.

40. The system of claim 39, wherein the first library of executable routines is one of embedded within the first CAD application and a component accessible by the first CAD application.

41. The system of claim 39, wherein the sub-set of native data results from utilization of at least one executable routine of the first library.

42. The system of claim 38, wherein the plug-in comprises a second library of executable routines and an API.

43. The system of claim 38, further comprising a third library of executable routines.

44. The system of claim 43, wherein the third library of executable routines is one of embedded within the second CAD application and a component accessible by the second CAD application.

45. The system of claim 43, wherein the third library of executable routines is suitable for reading the sub-set of native data utilizing the third library of executable routines to translate the sub-set of native data and create the second model of at least a portion of the object in the second CAD application.

46. In at least one electronic device, a method of communicating between a first CAD application and a second CAD application, comprising:

providing a first library of executable routines accessible by the first CAD application;  
generating native data as an object is modeled in the first CAD application;  
creating a sub-set of native data utilizing the first library of executable routines;  
storing the native data and the sub-set of native data;  
providing a second library of executable routines in the form of a plug-in having an application program interface (API) and being accessible by the second CAD application; and  
calling the second library of executable routines to utilize the API to retrieve the native data, filter the native data to extract the sub-set of native data, and convey the sub-set of native data to the second CAD application.

47. The method of claim 46, further comprising providing a third library of executable routines accessible by the second CAD application.

48. The method of claim 47, further comprising reading the sub-set of native data utilizing the third library of executable routines to translate the sub-set of native data and create a model of at least a portion of the object in the second CAD application.

49. In at least one electronic device, a method of communicating between a first application and a second application, comprising:

storing native data and a sub-set of native data;  
providing a plug-in having an API and being accessible by the second application;

and  
the plug-in in the sub-set of native data to the plug-in.

50. A computer readable medium containing software suitable for executing a method of communicating between a first computer aided design (CAD) application and a second CAD application, the method comprising:

storing native data and a sub-set of native data;

5 providing a plug-in having an application program interface (API) and being accessible by the second CAD application; and

the plug-in conveying the sub-set of native data to the second CAD application.

51. A computer readable medium containing software suitable for executing a method of communicating between a first CAD application and a second CAD application, the method comprising:

providing a model of an object on the first CAD application;

utilizing a plug-in in the second CAD application to retrieve a sub-set of native data from the first CAD application relating to the model of the object;

wherein the sub-set of native data enables the second CAD application to create a second model of at least a portion of the object on the second CAD application.

52. A computer readable medium containing software suitable for executing a method of communicating between a first CAD application and a second CAD application, the method comprising:

providing a first library of executable routines accessible by the first CAD application;

generating native data as an object is modeled in the first CAD application;

creating a sub-set of native data utilizing the first library of executable routines;

25 storing the native data and the sub-set of native data;

providing a second library of executable routines in the form of a plug-in having an application program interface (API) and being accessible by the second CAD application; and

calling the second library of executable routines to utilize the API to retrieve the native data, filter the native data to extract the sub-set of native data, and convey the sub-set of native data to the second CAD application.

53. A computer readable medium containing software suitable for executing a method of communicating between a first application and a second application, the method comprising:

storing native data and a sub-set of native data;

providing a plug-in having an API and being accessible by the second application;

5 and

the plug-in conveying the sub-set of native data to second CAD application.